Validation de modèle dynamique par corrélation calcul-essais

Résumé

Cette documentation est destinée à décrire les principaux outils d'aide à la validation de modèles en dynamique des structures par corrélation calcul-essais. On décrit notamment :
- comment importer des données issues de mesures,
- la validation par critère de MAC,
- la validation par comparaison de FRF calculées/simulées

Table des Matières

1 Introduction

Afin d'évaluer la capacité prédictive d'un modèle numérique en dynamique des structures, il peut être utile, voire indispensable de valider celui-ci par rapport à des données mesurées *in situ* . La manière la plus classique est la comparaison des modes propres (fréquences et déformées) calculés et identifiés expérimentalement deux à deux. Les modes propres reflètent le comportement global de la structure, et sont souvent utilisées comme unique outil de validation. La principale difficulté dans ce cas est d'être capable d'apparier les modes numériques et expérimentaux deux à deux.

Cette documentation a pour objectif de décrire les méthodes et outils utilisables dans Code_Aster pour comparer les données du calcul et de la mesure. On traite en section 2 de l'importation des données issues de logiciels de mesure dans *Code_Aster*. En section 3, on détaille l'utilisation d'un des principaux critères pour la validation de modèle : la matrice de MAC. En section 4, on aborde d'autres moyens proposés dans l'environnement de *Code_Aster* pour la validation (comparaison de FRF et de déformées).

2 Importer des données mesurées dans Code_Aster

2.1 Quelles données importer ?

Les données mesurées sont issues d'un logiciel d'acquisition et de traitement du signal. On peut citer parmi eux :

- LMS TestLab,
- Me'Scope,
- B&K Pulse,
- Labview,
- …

La plupart de ces logiciels permettent d'exporter des données au format universel, mis a point par le logiciel IDEAS (extension *.unv), données qui peuvent être relues dans *Code_Aster* par `LIRE_RESU(FORMAT='IDEAS')` .

Ces fichiers contiennent en général les informations relatives au maillage de la structure et aux données expérimentales. Les fichiers unv sont des fichiers ascii. Chaque ensemble de données est appelé « dataset », et est encadré dans le fichier par deux « -1 ». Le nombre qui suit la première occurrence « -1 » correspond au type de dataset. Chaque dataset est composé de plusieurs lignes (record), et chaque ligne contient des données rangées en colonnes (field)
Dans l'exemple ci-dessous, on présente quelques lignes d'un dataset 55, qui décrit une base de modes propres.

```
    -1
    55   %VALEURS AUX NOEUDS
ASTER  7.03.29 CONCEPT MODINTS1 CALC - CHAMP AUX NOEUDS DE NOM SYMBOLIQUE
CHAMP AUX NOEUDS DE NOM SYMBOLIQUE  DEPL - DX DY DZ DRX DRY DRZ
ASTER  7.03.29 CONCEPT MODINTS1 CALCULE LE 22/11/2004 A 19:24:57 DE TYPE
CHAMP AUX NOEUDS DE NOM SYMBOLIQUE  DEPL
NUMERO D'ORDRE: 1 NUME_MODE: 1 FREQ:  2.66902E-01
         1         2         3         8         2         6
         2         4         1         1
 2.18331e+01  1.00000e+00  3.74657e-03
      1011     % NOEUD NO1011
-1.37933E-001 7.39432E-007 3.38287E-001 0.00000E+000 0.00000E+000 0.00000E+000
      1001     % NOEUD NO1001
-1.37933E-001-2.80459E-009 1.72767E-001 0.00000E+000 0.00000E+000 0.00000E+000
    -1
```

2.1.1 Le maillage

Plusieurs data sets sont utilisés par les logiciels de mesure pour décrire un maillage. Celui présenté ici est le format utilisé par LMS pour exporter des maillages simples, composés uniquement de nœuds et de lignes les reliant. Les nœuds sont décrits par le data set 2411, et les connectivités par le data set 82 (exemple ci-dessous).

```
   -1
  2411
         1         0         0         8
   0.000000000000000e+000   0.000000000000000e+000   0.000000000000000e+000
         2         0         0         8
   0.000000000000000e+000   1.199999973177910e-001   0.000000000000000e+000
         3         0         0         8
   0.000000000000000e+000   3.300000131130219e-001   0.000000000000000e+000

    -1
    -1
    82
         1        30         8
LDN
         1         2         3         0        0         0         0
    -1
```

L'importation du maillage se fait dans Code_Aster avec `PRE_IDEAS` . La description des data sets 2411 et 82 est détaillée en annexe 5.1 .

2.1.2 Les données temporelles brutes, et FRF

Il est possible d'importer les données temporelles brutes, ou des FRF dans *Code_Aster*, afin de les comparer à des données simulées. Ces données sont stockées dans les fichiers unv sous le data set 58. On donne ci-dessous un exemple de ce type de data set :

```
     -1
    58
FRF (H1-estimator)
response / load
95-Oct-12 11:52:48
Alternate/identified FRF"
NONE
    0         4    0       139       NONE      1011    3       NONE        12  -3
         5      3124         1 4.00390e+000 1.95312e-002 0.00000e+000
        18    0    0    0 Frequency           Hz
        12    1    0    0 Channel 3(3)        m/s2
        13    0    1    0 Channel 1(1)        N
         0    0    0    0 Unknown             NONE
-1.33781e-001-5.07456e-003-1.31790e-001-5.19607e-003-1.29762e-001-5.32069e-003
-1.27699e-001-5.44850e-003-1.25597e-001-5.57962e-003-1.23458e-001-5.71414e-003
-1.21279e-001-5.85217e-003-1.19060e-001-5.99383e-003-1.16800e-001-6.13924e-003
-1.14497e-001-6.28851e-003-1.12151e-001-6.44177e-003-1.09760e-001-6.59917e-003
-1.07324e-001-6.76083e-003-1.04841e-001-6.92691e-003-1.02310e-001-7.09754e-003
```

Les headers décrivent le type de données. Ici, il s'agit d'une FRF « accélération/force », sur le degré de liberté $1011:+Z$ par rapport à la référence $12:-Z$.
Pour plus de détails, voir la documentation de référence du data set 58 en annexe 5.2 .

L'utilisation de `LIRE_RESU` dans ce cas ne pose pas de problèmes particuliers, et est décrit dans le cas-test sdls112a. A noter :

- les structures de données créées (temporelles ou fréquentielles) ne sont remplies que pour les DDL correspondant aux données lues. Cela peut engendrer des structures de données incomplètes, contrairement au $data\,set\,55$ dans lequel tous les degrés de liberté ont une valeur définie,
- la structure de données définie utilise un système de coordonnées local, avec les composantes `D1` , `D2` , et `D3` , dont les orientations sont données dans les composantes `D1X` , `D1Y` , `D1Z` , `D2X` , `D2Y` ... Il n'est pas possible d'afficher le résultat dans Salomé.
- Pour plus de détails, voir la documentation de `LIRE_RESU` .

2.1.3 Les bases de modes identifiés

Les modes identifiés sont stockés dans le $dataset\,55$, qui est dédié aux champs aux nœuds. Dans le $dataset\,58$, chaque bloc correspond à une fonction sur un nœud (équivalent à une fonction dans Aster), tandis que dans le $dataset\,55$, chaque bloc correspond à un champ aux nœuds (un `cham_no` dans Aster).

```
    -1
    55   %VALEURS AUX NOEUDS
ASTER  7.03.29 CONCEPT MODINTS1 CALC - CHAMP AUX NOEUDS DE NOM SYMBOLIQUE
CHAMP AUX NOEUDS DE NOM SYMBOLIQUE  DEPL - DX DY DZ DRX DRY DRZ
ASTER  7.03.29 CONCEPT MODINTS1 CALCULE LE 22/11/2004 A 19:24:57 DE TYPE
CHAMP AUX NOEUDS DE NOM SYMBOLIQUE  DEPL
NUMERO D'ORDRE: 1 NUME_MODE: 1 FREQ:  2.18331E+01
         1         2         2         8         2         3
         2         4         1         1
 2.18331e+01  1.00000e+00  3.74657e-03
      1011     % NOEUD NO1011
-1.37933E-001 7.39432E-007 3.38287E-001
      1001     % NOEUD NO1001
-1.37933E-001-2.80459E-009 1.72767E-001
    -1
```

La documentation détaillée du data set est donnée en annexe 5.3 .
Il est important de comprendre certaines caractéristiques de ce stockage, car les données doivent être rappelées à l'appel de `LIRE_RESU` . Notamment :

- Record 6 $(1,2,2,8,2,3)$:
- 1 : domaine de la mécanique des structures,
- 2 : on décrit un mode propre (« normal mode »)
- 2 : 3 degrés de liberté par noeud,
- 8 : champ de déplacement,
- 2 : champ réel (5 pour complexe),
- 6 : nombre de colonnes de valeurs
- Record 7 $(2,4,1,1)$:
- 2,4 : spécifique aux modes propres,
- 1 : cas de charge (1 par défaut),
- 1, numéro du mode
- Record 8 : $2.18331e+01\ 1.00000e+00\ 3.74657e-03\ 0.00000e+00$
- $2.18331e+01$: fréquence propre,
- $1.00000e+00$: masse modale,
- $3.74657e-03$: amortissement modal

La valeur de ces lignes est donnée dans `LIRE_RESU`, comme le montre l'exemple ci-dessous. Cela permet notamment à Aster de différencier dans un fichier unv les modes propres classiques des résidus statiques qui sont souvent calculés par les logiciels utilisés.

```
MODMES=LIRE_RESU(TYPE_RESU='MODE_MECA',
                 FORMAT='IDEAS',
                 MODELE=MODEXP,
                 UNITE=21,
                 NOM_CHAM='DEPL',
                 MATR_RIGI =KASSEXP,
                 MATR_MASS =MASSEXP,
                 FORMAT_IDEAS=_F(NOM_CHAM='DEPL',
                                 NUME_DATASET=55,
                                 RECORD_6=(1,2,2,8,2,3,),
                                 POSI_ORDRE=(7,4,),
                                 POSI_NUME_MODE=(7,4),
                                 POSI_FREQ=(8,1,),
                                 POSI_MASS_GENE=(8,2),
                                 POSI_AMOR_GENE=(8,3),
                                 NOM_CMP=('DX','DY','DZ'),),
                 TOUT_ORDRE='OUI',) ;
```

2.2 Création d'un modèle expérimental dans Code_Aster

La manipulation de données expérimentales dans Code_Aster nécessite de créer les structures de données adéquates, avec le formalisme du code. On doit donc reproduire toutes les étapes de la création du modèle, jusqu'à l'assemblage des matrices qui sont utilisées dans `LIRE_RESU` (mots-clés `MATR_RIGI` et `MATR_MASS` dans l'exemple ci-dessus).

2.2.1 Cas général

Dans le cas général, les nœuds sont tous reliés les uns aux autres par des éléments SEG2 linéaires. Les commandes à enchaîner sont les suivantes :

- importation et lecture du maillage avec `PRE_IDEAS` et `LIRE_MAILLAGE` ,
- affectation d'une modélisation mécanique de type `DIS_T` ; on pourrait utiliser une modélisation `DIS_TR` dans le cas où le champ à lire aurait 6 degrés de liberté par nœuds (par exemple, si on est capable de mesurer les degrés de liberté de rotation),
- affectation de caractéristiques géométriques en masse et raideur arbitraires sur les nœuds et segments avec `AFFE_CARA_ELEM` ,
- assemblage des matrices avec `ASSEMBLAGE` ,
- lecture des données avec `LIRE_RESU` .

Exemple : cas-test sdls112a, légèrement modifié afin de ne pas prendre en compte les nœuds orphelins.

```
PRE_IDEAS(UNITE_IDEAS=32,UNITE_MAILLAGE=22);

MAYAEXP=LIRE_MAILLAGE(UNITE=22);

MAYAEXP=DEFI_GROUP(reuse =MAYAEXP,
                   MAILLAGE=MAYAEXP,
                   CREA_GROUP_MA=_F(NOM='ALL_EXP',TOUT='OUI',),
                   CREA_GROUP_NO=_F(GROUP_MA='ALL_EXP',),);

MODEXP=AFFE_MODELE(MAILLAGE=MAYAEXP,
                   AFFE=_F(GROUP_MA='ALL_EXP',
                           PHENOMENE='MECANIQUE',
                           MODELISATION='DIS_T',),);

CHCAREXP=AFFE_CARA_ELEM(MODELE=MODEXP,
                        DISCRET=(_F(GROUP_MA='ALL_EXP',
                                    CARA='K_T_D_L',
                                    VALE=(1.0,1.0,1.0,),),
                                 _F(GROUP_MA='ALL_EXP',
                                    REPERE='GLOBAL',
                                    CARA='M_T_D_L',
                                    VALE=(1.0,),),),);

ASSEMBLAGE(     MODELE=MODEXP,
                CARA_ELEM=CHCAREXP,
                NUME_DDL=CO('NUMEXP'),
                MATR_ASSE=(_F(MATRICE=CO('KASSEXP'),
                              OPTION='RIGI_MECA',),
                           _F(MATRICE=CO('MASSEXP'),
                              OPTION='MASS_MECA',),),);

MODMES=LIRE_RESU(TYPE_RESU='MODE_MECA',
                 FORMAT='IDEAS',
                 MODELE=MODEXP,
                 UNITE=32,
                 NOM_CHAM='DEPL',
                 MATR_RIGI =KASSEXP,
                 MATR_MASS =MASSEXP,
                 FORMAT_IDEAS=_F(NOM_CHAM='DEPL',
                                 NUME_DATASET=55,
                                 RECORD_6=(1,2,2,8,2,3),
                                 POSI_ORDRE=(7,4,),
                                 POSI_NUME_MODE=(7,4),
                                 POSI_FREQ=(8,1,),
                                 POSI_MASS_GENE=(8,2),
                                 POSI_AMOR_GENE=(8,3),
                                 NOM_CMP=('DX','DY','DZ'),),
                 TOUT_ORDRE='OUI',);
```

Quelques conseils et pièges à éviter :

- L'ordre des composantes (mot-clé `NOM_CMP`) dans `LIRE_RESU` n'est pas obligatoire ; il est possible de faire un changement de repère simple (semblable pour tous les nœuds) en choisissant judicieusement l'ordre des composantes.
- Attention au mot-clé `RECORD_6` , celui-ci peut varier ; c'est notamment le cas lorsque le fichier unv a été créé par *Code_Aster* lui-même. En effet, il peut arriver que les données soient imprimées sur 6 colonnes, dans le cas où l'utilisateur a imprimé ses données sur une modélisation `DIS_TR` . Les 3 dernières colonnes contiennent les degrés de liberté de rotation. C'est le cas dans le cas-test sdls112a. On peut ne relire que les trois premières si le modèle expérimental ne possède que 3 degrés de liberté par nœud.
- Il est possible de lire des données issues de jauges de déformation. Dans ce cas,
- `NOM_CHAM='EPSI_NOEU'` ,
- `NOM_CMP=('EPXX','EPYY'...)` à choisir selon le repère utilisé.
- Pour comparer ces données à des données numériques, on pourra utiliser la macro-commande `OBSERVATION` .

2.2.2 Cas des nœuds orphelins

Il n'est pas conseillé d'utiliser des nœuds orphelins dans le maillage, car les champs associés sont difficiles à visualiser dans Salomé. On peut néanmoins lire des données expérimentales sur ces nœuds, à condition de leur appliquer une modélisation spécifique de type ponctuelle (`POI1`).
Le cas est traité dans le cas-test sdls112a.

```
MAYAEXP=CREA_MAILLAGE(MAILLAGE=MAYAtmp,
                      CREA_POI1=_F(TOUT='OUI',NOM_GROUP_MA='NOEU'),)


MODEXP=AFFE_MODELE(MAILLAGE=MAYAEXP,
                   AFFE=_F(GROUP_MA='NOEU',
                           PHENOMENE='MECANIQUE',
                           MODELISATION='DIS_T',),);

CHCAREXP=AFFE_CARA_ELEM(MODELE=MODEXP,
                        DISCRET=(_F(GROUP_MA='NOEU',
                                    REPERE='GLOBAL',
                                    CARA='K_T_D_N',
                                    VALE=(1.0,1.0,1.0,),),
                                 _F(GROUP_MA='NOEU',
                                    REPERE='GLOBAL',
                                    CARA='M_T_D_N',
                                    VALE=(1.0,),),),),) ;
```

3 Validation de modèle par critère de MAC

3.1 Qu'est-ce que le MAC ?

Le MAC, Modal Assurance Criterion, est un critère compris entre 0 et 1 donnant la colinéarité entre deux modes par rapport à une norme donnée.

$$MAC_{ij}=\frac{\left(\Phi_i^H W \Phi_j\right)^2}{\left(\Phi_i^H W \Phi_i\right)\left(\Phi_j^H W \Phi_j\right)}$$

L'utilisation des matrices de pondération (W dans la formule) est facultative. Lorsqu'on les connait, on peut utiliser les matrices de masse ou de raideur du modèle. C'est le cas lorsqu'on manipule des données numériques, car les matrices ont été assemblées sur le modèle. Elle permet de vérifier l'orthogonalité des modes propres par rapport aux matrices de masse et de raideur :

- $MAC_{ij}=1\,si\;i=j$
- $MAC_{ij}=0\,sinon$

Mais lorsqu'on manipule des données expérimentales, on ne connait pas les matrices condensées sur ce modèle. On peut les fabriquer par condensation de Guyan à partir du modèle numérique, mais celui-ci n'étant pas recalé, on risque de commettre une erreur.

On peut, plus simplement, calculer le MAC sans matrice de pondération, et regarder la colinéarité des modes sur la norme L_2 .

- Si l'objectif est de vérifier l'orthogonalité de la base, on peut considérer, en première approximation, que le MAC sans matrice de pondération est assez semblable au MAC pondéré par la matrice de masse,
- Si l'objectif est de comparer deux bases de modes entre elles, alors, le choix de cette norme est équivalent aux autres : le MAC vaudra 1 si les modes sont colinéaires (donc s'ils « se ressemblent ») et 0 sinon.

NB : l'utilisation du MAC sur des modes expérimentaux permet notamment de vérifier la capacité des capteurs à séparer les modes. En effet, plus on a de capteurs, plus les modes « auront l'air différents » vus de ceux-ci. Le MAC de deux modes différents correctement identifiés sera donc proche de 0. Si on a un seul capteur, alors le MAC entre deux modes vaudra toujours 1 : les modes ne sont pas séparables.

3.2 Projection de champs

3.2.1 Projection des données numériques sur le modèle expérimental avec PROJ_CHAMP

Les modes ne sont comparables que s'ils sont définis sur le même modèle. On projette donc la base de modes numérique, calculée avec *Code_Aster* , sur le modèle expérimental, avec la commande `PROJ_CHAMP` .

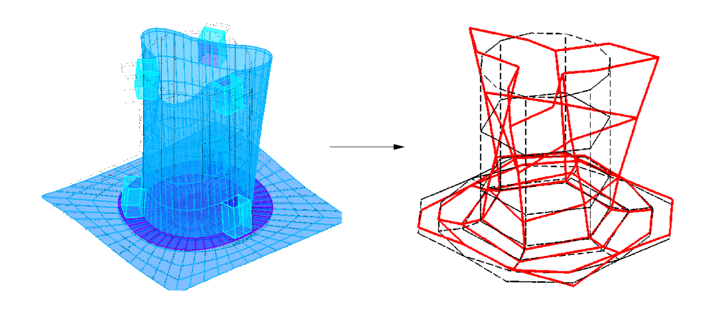


Image 3.2-1: projection de données.

Remarques :

- Note importante : il faut préciser, dans `PROJ_CHAMP` , le nom du `NUME_DDL` du modèle expérimental, de manière à ce que les numérotations des modes expérimentaux et des modes numériques projetés soient les mêmes.
- Si le `NUME_DDL` n'est pas le même, on ne pourra pas calculer de critère de MAC.
- Dans `PROJ_CHAMP` , il est nécessaire de spécifier la dimension sur laquelle on projette : par défaut, on va associer les nœuds du modèle expérimental à des éléments 3D du modèle numérique.
- Si le modèle numérique est constitué d'éléments de plaque, il faut le préciser avec `CAS='2.5D'` ,
- Si le modèle numérique est composé d'éléments 3D et 2D, alors il n'est pas possible de spécifier plusieurs types de projections. Une solution proposée est d'affecter une modélisation de plaque (`DKT` par exemple) aux éléments de peau qui recouvrent les éléments 3D et de se placer dans le cas `'2.5D'` .
- L'opérateur `OBSERVATION` permet de réaliser la même opération, avec des options supplémentaires :
- utilisation de repères locaux capteurs par capteurs,
- suppression des données mesurées de la structure de données résultat (pour les cas où la mesure avait été faite avec des capteurs uni-axiaux),
- création d'une structure de données mixtes comprenant des données accélérométriques et extensiométriques, pour reproduire une mesure accéléromètres + jauges,
- simulation d'une « jauge virtuelle ».
- La description de cet opérateur est proposée dans le paragraphe suivant.

3.2.2 Utilisation de la macro-commande OBSERVATION pour le projection des données

On propose de donner un exemple pratique de l'utilisation de la macro-commande sur le cas suivant :

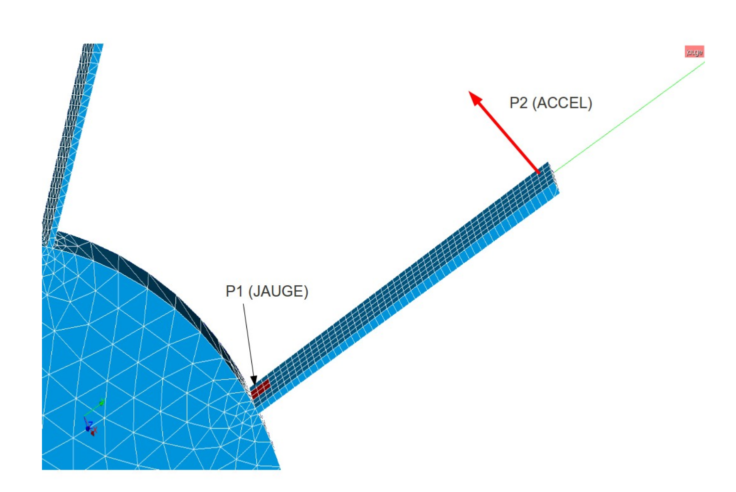


Image 3.2-2: mesure des vibrations d'une ailette à l'aide d'une jauge et d'un accéléromètre.

On suppose avoir mesuré les vibrations de la roue aubagée représentée sur le maillage ci-dessous en posant une jauge à la base de chaque ailette et un accéléromètre au sommet. Les modes propres identifiés sont exportés au format unv **sans passer au repère global**. Les vibrations au sommet n'ayant été mesurées que dans une direction, on aura un ligne de la forme :

```
     2          4          1          1
2.18331e+01  1.00000e+00  3.74657e-03
     1          % NOEUD NO1
0.00000e+00  0.00000e+00  7.39432e-01
   -1
```

Sur le nœud $N01$, seule la composante locale DZ a été mesurée. Les directions non mesurées sont mises à 0. Il n'est pas possible, dans un data set 55, de faire la différence entre données non mesurées et mesures nulles.

Pour comparer les données expérimentales et les modes propres numériques réduits, deux opérations sont nécessaires :

- sur les données expérimentales, on filtre les modes de manière à éliminer les données non mesurées, en ne gardant qu'une direction pour la jauge et l'accéléromètre
- on utilise OBSERVATION sans projection (PROJECTION='NON'), car on ne fait que filtrer les données,

```
OBSMXT = OBSERVATION(  RESULTAT = MESURE,
                       MODELE_1 = MODMESUR,
                       MODELE_2 = MODMESUR,
                       PROJECTION = 'NON',
                       TOUT_ORDRE = 'OUI',
                       NOM_CHAM = ('DEPL','EPSI_NOEU',),
                       FILTRE =(   _F( GROUP_NO = 'P1',
                                       NOM_CHAM = 'EPSI_NOEU',
                                       DDL_ACTIF = ('EPXX',),),
                                   _F( GROUP_NO = 'P2',
                                       NOM_CHAM = 'DEPL',
                                       DDL_ACTIF = ('DZ',),), ),);
```

- sur les données numériques, on projette les modes propres sur les modèle expérimental :
- on utilise PROJECTION='OUI' ,
- on calcule la déformation moyenne pour le groupe de nœuds en rouge sur la figure avant de réaliser la projection : cette surface correspond à la surface effectivement mesurée par la jauge,

- on fait les changements de repère, en utilisant l'option `'NORMALE'` : on calcule la normale au maillage numérique pour définir l'axe Z du repère local (le second axe est défini avec le mot-clé `VECT_Y`) ; ici, on aurait pu aussi utiliser l'option `'CYNLINDRIQUE'` .
- on filtre les composantes correspondant aux données mesurées.

```
OBSJAU = OBSERVATION( RESULTAT = CALCUL,
                      MODELE_1 = MODNUME,
                      MODELE_2 = MODMESUR,
                      PROJECTION = 'OUI',
                      TOUT_ORDRE = 'OUI',
                      NOM_CHAM = 'EPSI_NOEU',
                      EPSI_MOYENNE =  _F( GROUP_MA='SURF1',SEUIL_VARI=(0.1,),
                                          MASQUE=('EPYY','EPZZ','EPXY',
                                                  'EPXZ','EPYZ'),),
                      MODI_REPERE = _F( GROUP_NO = ('P1',P2'),
                                        REPERE = 'NORMALE',
                                        VECT_Y = (0.,1.,0.)),),
                      FILTRE =( _F( GROUP_NO = 'P1',
                                    NOM_CHAM = 'EPSI_NOEU',
                                    DDL_ACTIF = ('EPXX'),),
                                _F( GROUP_NO = 'P2',
                                    NOM_CHAM = 'DEPL',
                                    DDL_ACTIF = ('DZ',),),),),) ;
```

3.3 Calcul du MAC entre deux bases de modes

Le calcul du MAC peut être réalisé avec l'opérateur `MAC_MODES`, qui calcule la matrice de MAC entre tous les modes de deux bases. La structure de données produite est une table, qu'on imprime avec `INFO=2` dans `MAC_MODES`.
Le mot-clé `MATR_ASSE` permet d'utiliser une matrice de pondération.
La table imprimée a la forme suivante :

```
 ASTER 11.01.03 CONCEPT MAC_ET CALCULE LE 15/02/2012 A 14:18:54 DE TYPE
  TABLE_SDASTER
MAC                        ! NUME_MODE_1
                           !            1             2             3             4             5
---------------------------------------------------------------------------------------------------
NUME_MODE_2              1 !   1.00000E+00  2.17692E-14  7.49505E-16  5.23742E-22  1.66188E-21
                         2 !   2.17692E-14  1.00000E+00  4.21440E-13  9.40269E-19  1.12652E-19
                         3 !   7.49505E-16  4.21440E-13  1.00000E+00  7.24387E-18  1.28403E-17
                         4 !   5.23742E-22  9.40269E-19  7.24387E-18  1.00000E+00  2.11012E-13
                         5 !   1.66188E-21  1.12652E-19  1.28403E-17  2.11012E-13  1.00000E+00
```

On peut visualiser le MAC produit dans excel, ou utiliser la macro-commande `CALC_ESSAI`, qui propose un visualisation en 2D. Pour cela, lancer la macro-commande sans mot-clés à la fin du calcul, et se positionner sur l'onglet « expansion de modèles ».

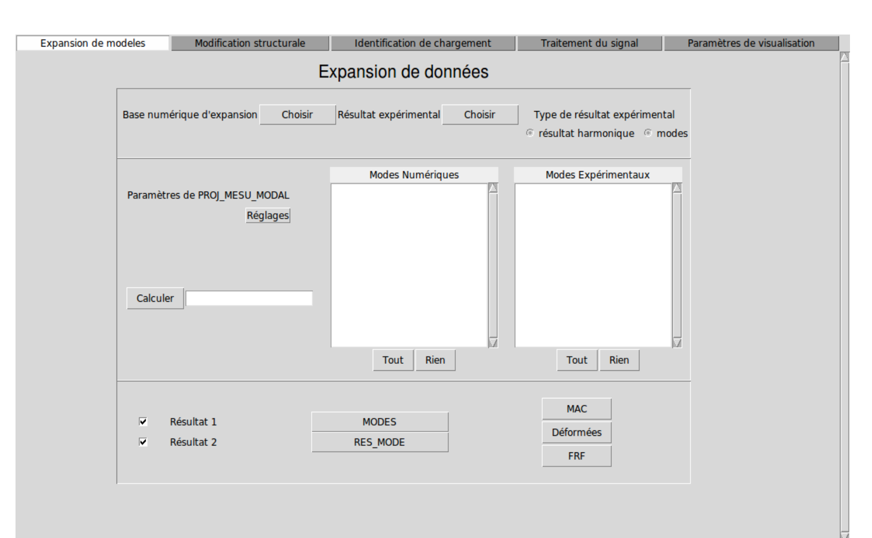


Image 3.3-1: Image 3.3-1: CALC_ESSAI : expansion de modèles.

Dans le cadre du bas, choisir les deux bases de modes à comparer (si une seule des deux bases est sélectionnée, on fera un MAC de la base par elle-même), et cliquer sur MAC. Si les bases choisies ne sont pas définies sur le même modèle, le bouton MAC est grisé.
La matrice de MAC qui apparaît est la suivante :

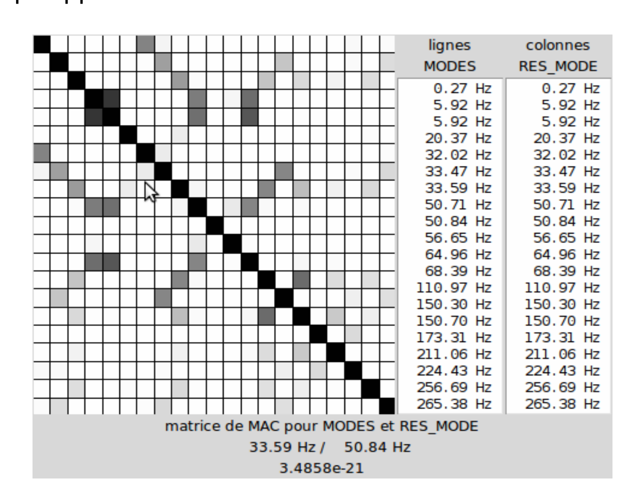


Image 3.3-2: MAC entre deux bases de modes visualisé sous CALC_ESSAI.

En passant la souris sur les cases, on voit en bas les fréquences des modes concernés et la valeur du MAC pour ces derniers.

NB : en annexe 5.4, on propose un script utilisant la bibliothèque python matplotlib permettant de créer des diagrammes de MAC en 3D plus aisément interprétables que celui implémenté par défaut dans `CALC_ESSAI`. A plus long terme, on étudie la faisabilité d'intégrer le MAC 3D par défaut à l'opérateur.

4 Autres méthodes de validation

4.1 Validation par comparaison visuelle de déformées modales

Ce mode de validation est le plus direct. Il peut se faire en imprimant de manière classique les déformées modales dans Salomé.

NB : dans `CALC_ESSAI`, dans l'onglet « Expansion de modèles », il est possible de sélectionner une ou deux bases (en face de « Résultat 1 » et « Résultat 2 ») et de les visualiser dans GMSH en cliquant sur « Déformées ». Il n'est pas possible actuellement de visualiser les déformées dans Salomé, ce développement doit être réalisé en 2012 (en ajoutant la possibilité de superposer les déformées).

4.2 Validation par comparaison de FRF

On propose une procédure dans CALC_ESSAI permettant de comparer une FRF issue de la mesure à une FRF simulée par coup de marteau. Cette méthode de validation est différente, car la comparaison se fait sur un point de mesure à la fois, mais sur une bande de fréquence étendue. Elle permet de vérifier la validité même du modèle modal.
Pour cela, cliquer sur « FRF » dans l'onglet « Expansion de modèles ». La fenêtre suivante apparaît :

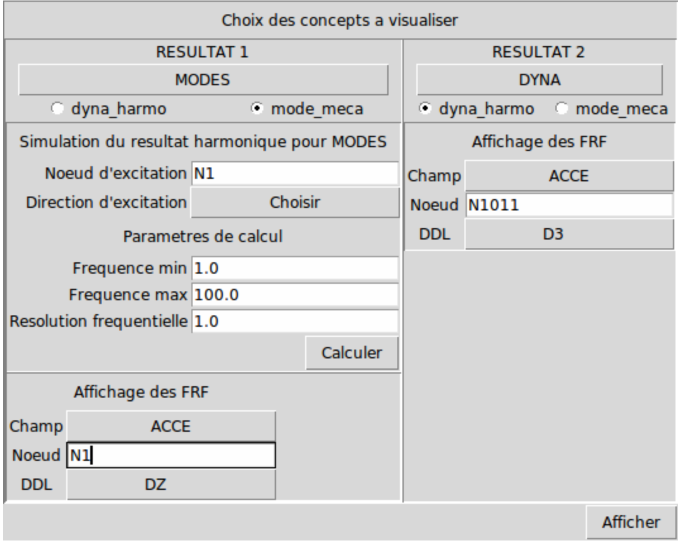


Image 4.2-1: simulations de FRF dans CALC_ESSAI.

On peut sélectionner d'un côté un concept de type modes et simuler une FRF, et visualiser de l'autre côté une FRF expérimentale mesurée. En affichant les courbes, on peut obtenir le graphe suivant, produit dans XMGrace :

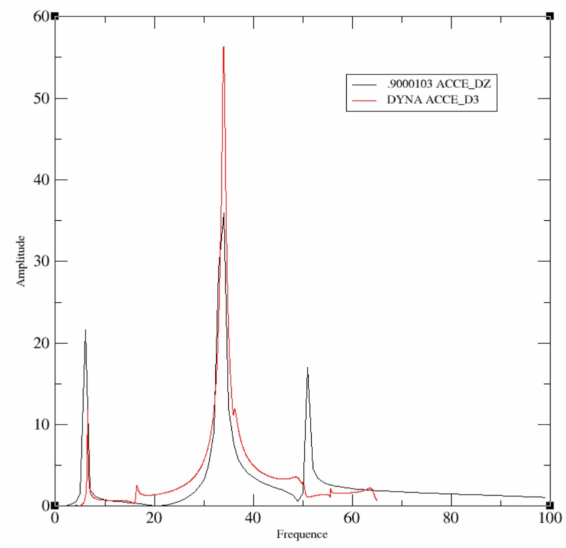


Image 4.2-2: affichage de FRF dans XMGrace.

5 Annexe

5.1 Documentation unv sur les data set de maillage

Data set 2411 : description des nœuds :

```
Name:   Nodes - Double Precision
Status: Current
Owner:  Simulation
Revision Date: 23-OCT-1992
----------------------------------------------------------------------------
 Record 1:        FORMAT(4I10)
                 Field 1       -- node label
                 Field 2       -- export coordinate system number
                 Field 3       -- displacement coordinate system number
                 Field 4       -- color
Record 2:        FORMAT(1P3D25.16)
                 Fields 1-3    -- node coordinates in the part coordinate
                                  system

Records 1 and 2 are repeated for each node in the model.

Example:

    -1
  2411
       121         1         1        11
   5.0000000000000000D+00   1.0000000000000000D+00   0.0000000000000000D+00
       122         1         1        11
   6.000000000
```

Data set 82 : description des connectivités : ce data set n'est plus utilisé que dans les cas très particuliers de maillages expérimentaux. Les éléments sont plus généralement décrits par le data set 2412.

```
Name:   Tracelines
Status: Obsolete
Owner:  Simulation
Revision Date: 27-Aug-1987
Additional Comments:  This dataset is written by I-DEAS Test.
-----------------------------------------------------------------------

             Record 1: FORMAT(3I10)
                       Field 1 -    trace line number
                       Field 2 -    number of nodes defining trace line
                                    (maximum of 250)
                       Field 3 -    color

             Record 2: FORMAT(80A1)
                       Field 1 -    Identification line

             Record 3: FORMAT(8I10)
                       Field 1 -    nodes defining trace line
                               =    > 0 draw line to node
                               =    0 move to node (a move to the first
                                    node is implied)
             Notes: 1) MODAL-PLUS node numbers must not exceed 8000.
                    2) Identification line may not be blank.
                    3) Systan only uses the first 60 characters of the
                       identification text.
                    4) MODAL-PLUS does not support trace lines longer than
                       125 nodes.
                    5) Supertab only uses the first 40 characters of the
                       identification line for a name.
                    6) Repeat Datasets for each Trace_Line
```

Data set 2412 : description des éléments (modèle EF classique) :

```
Name:   Elements
Status: Current
Owner:  Simulation
Revision Date: 14-AUG-1992
-----------------------------------------------------------------------

Record 1:        FORMAT(6I10)
                 Field 1       -- element label
                 Field 2       -- fe descriptor id
                 Field 3       -- physical property table number
                 Field 4       -- material property table number
                 Field 5       -- color
                 Field 6       -- number of nodes on element

Record 2:  *** FOR NON-BEAM ELEMENTS ***
                 FORMAT(8I10)
                 Fields 1-n    -- node labels defining element

Record 2:  *** FOR BEAM ELEMENTS ONLY ***
                 FORMAT(3I10)
                 Field 1       -- beam orientation node number
                 Field 2       -- beam fore-end cross section number
                 Field 3       -- beam  aft-end cross section number

Record 3:  *** FOR BEAM ELEMENTS ONLY ***
                 FORMAT(8I10)
                 Fields 1-n    -- node labels defining element

Records 1 and 2 are repeated for each non-beam element in the model.
Records 1 - 3 are repeated for each beam element in the model.

Example:

    -1
  2412
         1        11         1      5380         7         2
         0         1         1
         1         2
         2        21         2      5380         7         2
         0         1         1
         3         4
         3        22         3      5380         7         2
         0         1         2
         5         6
         6        91         6      5380         7         3
        11        18        12
         9        95         6      5380         7         8
        22        25        29        30        31        26        24        23
        14       136         8         0         7         2
        53        54
        36       116        16      5380         7        20
       152       159       168       167       166       158       150       151
       154       170       169       153       157       161       173       172
       171       160       155       156
     -1
```

5.2 Documentation de référence sur le data set 58

```
Number: 58
Name:   Function at Nodal DOF
Status: Current
Owner:  Test
Revision Date: 23-Apr-1993
-----------------------------------------------------------------------

         Record 1:     Format(80A1)
                       Field 1    - ID Line 1

                                                   NOTE

                           ID Line 1 is generally  used  for  the  function
                           description.


         Record 2:     Format(80A1)
```

```
              Field 1    - ID Line 2

Record 3:     Format(80A1)
              Field 1    - ID Line 3

                                    NOTE

                  ID Line 3 is generally used to identify when the
                  function  was  created.  The date is in the form
                  DD-MMM-YY, and the time is in the form HH:MM:SS,
                  with a general Format(9A1,1X,8A1).


Record 4:     Format(80A1)
              Field 1    - ID Line 4

Record 5:     Format(80A1)
              Field 1    - ID Line 5

Record 6:     Format(2(I5,I10),2(1X,10A1,I10,I4))
                         DOF Identification
              Field 1    - Function Type
                           0 - General or Unknown
                           1 - Time Response
                           2 - Auto Spectrum
                           3 - Cross Spectrum
                           4 - Frequency Response Function
                           5 - Transmissibility
                           6 - Coherence
                           7 - Auto Correlation
                           8 - Cross Correlation
                           9 - Power Spectral Density (PSD)
                           10 - Energy Spectral Density (ESD)
                           11 - Probability Density Function
                           12 - Spectrum
                           13 - Cumulative Frequency Distribution
                           14 - Peaks Valley
                           15 - Stress/Cycles
                           16 - Strain/Cycles
                           17 - Orbit
                           18 - Mode Indicator Function
                           19 - Force Pattern
                           20 - Partial Power
                           21 - Partial Coherence
                           22 - Eigenvalue
                           23 - Eigenvector
                           24 - Shock Response Spectrum
                           25 - Finite Impulse Response Filter
                           26 - Multiple Coherence
                           27 - Order Function
              Field 2    - Function Identification Number
              Field 3    - Version Number, or sequence number
              Field 4    - Load Case Identification Number
                           0 - Single Point Excitation
              Field 5    - Response Entity Name ("NONE" if unused)
              Field 6    - Response Node
              Field 7    - Response Direction
                            0 - Scalar
                            1 - +X Translation       4 - +X Rotation
                           -1 - -X Translation      -4 - -X Rotation
                            2 - +Y Translation       5 - +Y Rotation
                           -2 - -Y Translation      -5 - -Y Rotation
                            3 - +Z Translation       6 - +Z Rotation
                           -3 - -Z Translation      -6 - -Z Rotation
              Field 8    - Reference Entity Name ("NONE" if unused)
              Field 9    - Reference Node
              Field 10   - Reference Direction  (same as field 7)

                                    NOTE

                  Fields 8, 9, and 10 are only relevant if field 4
                  is zero.


Record 7:     Format(3I10,3E13.5)
                         Data Form
              Field 1    - Ordinate Data Type
```

```
                         2 - real, single precision
                         4 - real, double precision
                         5 - complex, single precision
                         6 - complex, double precision
            Field 2    - Number of data pairs for uneven  abscissa
                         spacing, or number of data values for even
                         abscissa spacing
            Field 3    - Abscissa Spacing
                         0 - uneven
                         1 - even (no abscissa values stored)
            Field 4    - Abscissa minimum (0.0 if spacing uneven)
            Field 5    - Abscissa increment (0.0 if spacing uneven)
            Field 6    - Z-axis value (0.0 if unused)

Record 8:   Format(I10,3I5,2(1X,20A1))
                       Abscissa Data Characteristics
            Field 1    - Specific Data Type
                         0 - unknown
                         1 - general
                         2 - stress
                         3 - strain
                         5 - temperature
                         6 - heat flux
                         8 - displacement
                         9 - reaction force
                         11 - velocity
                         12 - acceleration
                         13 - excitation force
                         15 - pressure
                         16 - mass
                         17 - time
                         18 - frequency
                         19 - rpm
                         20 - order
            Field 2    - Length units exponent
            Field 3    - Force units exponent
            Field 4    - Temperature units exponent

                                    NOTE

                Fields 2, 3 and  4  are  relevant  only  if  the
                Specific Data Type is General, or in the case of
                ordinates, the response/reference direction is a
                scalar, or the functions are being used for
                nonlinear connectors in System Dynamics Analysis.
                See Addendum 'A' for the units exponent table.

            Field 5    - Axis label ("NONE" if not used)
            Field 6    - Axis units label ("NONE" if not used)

                                    NOTE

                If fields  5  and  6  are  supplied,  they  take
                precendence  over  program  generated labels and
                units.

Record 9:   Format(I10,3I5,2(1X,20A1))
            Ordinate (or ordinate numerator) Data Characteristics

Record 10:  Format(I10,3I5,2(1X,20A1))
            Ordinate Denominator Data Characteristics

Record 11:  Format(I10,3I5,2(1X,20A1))
            Z-axis Data Characteristics

                                    NOTE

                Records 9, 10, and 11 are  always  included  and
                have fields the same as record 8.  If records 10
                and 11 are not used, set field 1 to zero.

Record 12:
                        Data Values

                 Ordinate            Abscissa
  Case     Type     Precision     Spacing       Format
-------------------------------------------------------------
```

```
          1      real      single        even         6E13.5
          2      real      single       uneven        6E13.5
          3     complex    single        even         6E13.5
          4     complex    single       uneven        6E13.5
          5      real      double        even         4E20.12
          6      real      double       uneven     2(E13.5,E20.12)
          7     complex    double        even         4E20.12
          8     complex    double       uneven      E13.5,2E20.12
      --------------------------------------------------------------

                                   NOTE

      See  Addendum  'B'  for  typical  FORTRAN   READ/WRITE
      statements for each case.


    General Notes:

         1.  ID lines may not be blank.  If no  information  is  required,
             the word "NONE" must appear in columns 1 through 4.

         2.  ID line 1 appears on plots in Finite Element Modeling and is
             used as the function description in System Dynamics Analysis.

         3.  Dataloaders use the following ID line conventions
                ID Line 1 - Model Identification
                ID Line 2 - Run Identification
                ID Line 3 - Run Date and Time
                ID Line 4 - Load Case Name

         4.  Coordinates codes from MODAL-PLUS and MODALX are decoded into
             node and direction.

         5.  Entity names used in System Dynamics Analysis prior to I-DEAS
             Level 5 have a 4 character maximum. Beginning with Level 5,
             entity names will be ignored if this dataset is preceded by
             dataset 259. If no dataset 259 precedes this dataset, then the
             entity name will be assumed to exist in model bin number 1.

         6.  Record 10 is ignored by System Dynamics Analysis unless load
             case = 0. Record 11 is always ignored by System Dynamics
             Analysis.

         7.  In record 6, if the response or reference names are "NONE"
             and are not overridden by a dataset 259, but the correspond-
             ing node is non-zero, System Dynamics Analysis adds the node
             and direction to the function description if space is sufficie

         8.  ID line 1 appears on XY plots in Test Data Analysis along
             with ID line 5 if it is defined.  If defined, the axis units
             labels also appear on the XY plot instead of the normal
             labeling based on the data type of the function.

         9.  For functions used with nonlinear connectors in System
             Dynamics Analysis, the following requirements must be
             adhered to:

             a) Record 6: For a displacement-dependent function, the
                function type must be 0; for a frequency-dependent
                function, it must be 4. In either case, the load case
                identification number must be 0.

             b) Record 8: For a displacement-dependent function, the
                specific data type must be 8 and the length units
                exponent must be 0 or 1; for a frequency-dependent
                function, the specific data type must be 18 and the
                length units exponent must be 0. In either case, the
                other units exponents must be 0.

             c) Record 9: The specific data type must be 13. The
                temperature units exponent must be 0. For an ordinate
                numerator of force, the length and force units
                exponents must be 0 and 1, respectively. For an
                ordinate numerator of moment, the length and force
                units exponents must be 1 and 1, respectively.

             d) Record 10: The specific data type must be 8 for
```

```
                      stiffness and hysteretic damping; it must be 11
                      for viscous damping. For an ordinate denominator of
                      translational displacement, the length units exponent
                      must be 1; for a rotational displacement, it must
                      be 0. The other units exponents must be 0.

                   e) Dataset 217 must precede each function in order to
                      define the function's usage (i.e. stiffness, viscous
                      damping, hysteretic damping).
```

5.3 Documentation de référence sur le data set 55

```
Name:   Data at Nodes
Status: Obsolete
Owner:  Simulation
Revision Date: 07-Mar-1997
Additional Comments: This dataset is written and read by I-DEAS Test.
-----------------------------------------------------------------------

          RECORD 1:      Format (40A2)
               FIELD 1:          ID Line 1

          RECORD 2:      Format (40A2)
               FIELD 1:          ID Line 2

          RECORD 3:      Format (40A2)

               FIELD 1:          ID Line 3

          RECORD 4:      Format (40A2)
               FIELD 1:          ID Line 4

          RECORD 5:      Format (40A2)
               FIELD 1:          ID Line 5

          RECORD 6:      Format (6I10)

          Data Definition Parameters

               FIELD 1: Model Type
                           0:   Unknown
                           1:   Structural
                           2:   Heat Transfer
                           3:   Fluid Flow

               FIELD 2:  Analysis Type
                           0:   Unknown
                           1:   Static
                           2:   Normal Mode
                           3:   Complex eigenvalue first order
                           4:   Transient
                           5:   Frequency Response
                           6:   Buckling
                           7:   Complex eigenvalue second order

               FIELD 3:  Data Characteristic
                           0:   Unknown
                           1:   Scalar
                           2:   3 DOF Global Translation
                                Vector
                           3:   6 DOF Global Translation
                                & Rotation Vector
                           4:   Symmetric Global Tensor
                           5:   General Global Tensor


               FIELD 4: Specific Data Type
                           0:   Unknown
                           1:   General
                           2:   Stress
                           3:   Strain (Engineering)
                           4:   Element Force
                           5:   Temperature
                           6:   Heat Flux
                           7:   Strain Energy
```

```
                    8:   Displacement
                    9:   Reaction Force
                    10:   Kinetic Energy
                    11:   Velocity
                    12:   Acceleration
                    13:   Strain Energy Density
                    14:   Kinetic Energy Density
                    15:   Hydro-Static Pressure
                    16:   Heat Gradient
                    17:   Code Checking Value
                    18:   Coefficient Of Pressure

        FIELD 5:  Data Type
                    2:   Real
                    5:   Complex

        FIELD 6:  Number Of Data Values Per Node (NDV)


   Records 7 And 8 Are Analysis Type Specific

   General Form

   RECORD 7:      Format (8I10)

        FIELD 1:          Number Of Integer Data Values
                    1 < Or = Nint < Or = 10
        FIELD 2:          Number Of Real Data Values
                    1 < Or = Nrval < Or = 12
        FIELDS 3-N:       Type Specific Integer Parameters


   RECORD 8:      Format (6E13.5)
        FIELDS 1-N:       Type Specific Real Parameters


   For Analysis Type = 0, Unknown

   RECORD 7:

        FIELD 1:   1
        FIELD 2:   1
        FIELD 3:   ID Number

   RECORD 8:

        FIELD 1:   0.0

   For Analysis Type = 1, Static

   RECORD 7:
        FIELD 1:    1
        FIELD 2:    1
        FIELD 3:    Load Case Number

   RECORD 8:
        FIELD 11:    0.0

   For Analysis Type = 2, Normal Mode


  RECORD 7:

        FIELD 1:    2
        FIELD 2:    4
        FIELD 3:    Load Case Number
        FIELD 4:    Mode Number

   RECORD 8:
        FIELD 1:    Frequency (Hertz)
        FIELD 2:    Modal Mass
        FIELD 3:    Modal Viscous Damping Ratio
        FIELD 4:    Modal Hysteretic Damping Ratio

   For Analysis Type = 3, Complex Eigenvalue

   RECORD 7:
```

```
          FIELD 1:    2
          FIELD 2:    6
          FIELD 3:    Load Case Number
          FIELD 4:    Mode Number

     RECORD 8:

          FIELD 1:    Real Part Eigenvalue
          FIELD 2:    Imaginary Part Eigenvalue
          FIELD 3:    Real Part Of Modal A
          FIELD 4:    Imaginary Part Of Modal A
          FIELD 5:    Real Part Of Modal B
          FIELD 6:    Imaginary Part Of Modal B


     For Analysis Type = 4, Transient

     RECORD 7:

          FIELD 1:    2
          FIELD 2:    1
          FIELD 3:    Load Case Number
          FIELD 4:    Time Step Number

     RECORD 8:
          FIELD 1: Time (Seconds)


     For Analysis Type = 5, Frequency Response

     RECORD 7:

          FIELD 1:    2
          FIELD 2:    1
          FIELD 3:    Load Case Number
          FIELD 4:    Frequency Step Number

     RECORD 8:
          FIELD 1:    Frequency (Hertz)

     For Analysis Type = 6, Buckling

     RECORD 7:

          FIELD 1:    1
          FIELD 2:    1
          FIELD 3:    Load Case Number

     RECORD 8:

          FIELD 1: Eigenvalue

     RECORD 9:      Format (I10)

          FIELD 1:          Node Number

     RECORD 10:     Format (6E13.5)
          FIELDS 1-N:       Data At This Node (NDV Real Or
                    Complex Values)

     Records 9 And 10 Are Repeated For Each Node.
```

5.4 Script pour la représentation 3D d'un diagramme de MAC

Ce script peut être recopié en bas d'un fichier de commande, en remplaçant les noms $B1$ et $B2$ sur la dernière ligne par les noms des deux bases que l'on souhaite comparer par MAC.

Attention : ce script s'appuie sur la bibliothèque matplotlib qui doit être installée, et qui n'est pas actuellement un pré-requis de *Code_Aster*. Il est prévu d'incorporer à `CALC_ESSAI` les fonctionnalités permettant d'arriver à ce résultat, mais ce n'est pas le cas aujourd'hui.

```
def mac_plot_lib(BASE1,BASE2):
    """ calcule le mac entre deux bases, l'extrait et le represente dans un graphe 3d
        matplotlib"""
    __MAC=MAC_MODES(BASE_1=BASE1,
                    BASE_2=BASE2);

    mactmp=__MAC.EXTR_TABLE()
    mac = mactmp.Croise()
    mac_py = mac.values()

    import numpy as np
    from mpl_toolkits.mplot3d import axes3d
    import matplotlib.pyplot as plt

    freq_1 = BASE1.LIST_PARA()['FREQ']
    freq_2 = BASE2.LIST_PARA()['FREQ']
    nume_ordre_1 = BASE1.LIST_PARA()['NUME_ORDRE']
    nume_ordre_2 = BASE2.LIST_PARA()['NUME_ORDRE']
    nb_freq_1 = len(freq_1)
    nb_freq_2 = len(freq_2)
    matrice_mac = np.transpose(np.array([ mac_py[kk] for kk in nume_ordre_1]))

    fig = plt.figure()
    ax = axes3d.Axes3D(fig)

    # Create regular mesh from coordinates
    xpos,ypos = np.meshgrid(np.arange(nb_freq_1),range(nb_freq_2))
    xpos = xpos + 0.5*(np.ones(matrice_mac.shape)-matrice_mac)
    ypos = ypos + 0.5*(np.ones(matrice_mac.shape)-matrice_mac)
    xpos = xpos.flatten()
    ypos = ypos.flatten()
    dx = matrice_mac.flatten()
    dy = dx.copy()
    dz = dx.copy()
    zpos=np.zeros(nb_freq_1*nb_freq_2)

    for kk in range(len(xpos)):
        if dx[kk]<1.0E-6:
            # pour eviter les plantages en cas de mac trop petit
            dx[kk]=dy[kk]=dz[kk]=1.0E-6
        ax.bar3d(xpos[kk],ypos[kk],zpos[kk],
                 dx[kk],dy[kk],dz[kk],
                 color=mac2col(dz[kk]))

    ax.set_xlabel(u'FREQ_I')
    ax.set_ylabel(u'FREQ_J')
    ax.set_zlabel(u'MAC')
    plt.show()


def mac2col(value):
    # donne la valeur de la couleur correspondant a une valeur de MAC
    # comprise entre 0 et 1
    import matplotlib.colors as colors
    import matplotlib.cm as cm
    value = 1-value
    desc=cm.RdYlBu._segmentdata
    segments=[desc['blue'][kk][0] for kk in range(len(desc['blue']))]
    num_seg=0
    for kk in segments:
        if value > kk:
            num_seg = num_seg+1
    tri=(desc['red'][num_seg][1],
         desc['green'][num_seg][1],
         desc['blue'][num_seg][1])
    return colors.rgb2hex(tri)


mac_plot_lib(B1,B2)
```

Type de résultat attendu : un problème de visualisation 3D avec matplotlib a été relevé, aucune correction n'est pour le moment prévue. Ce problème n'est pas très gênant.

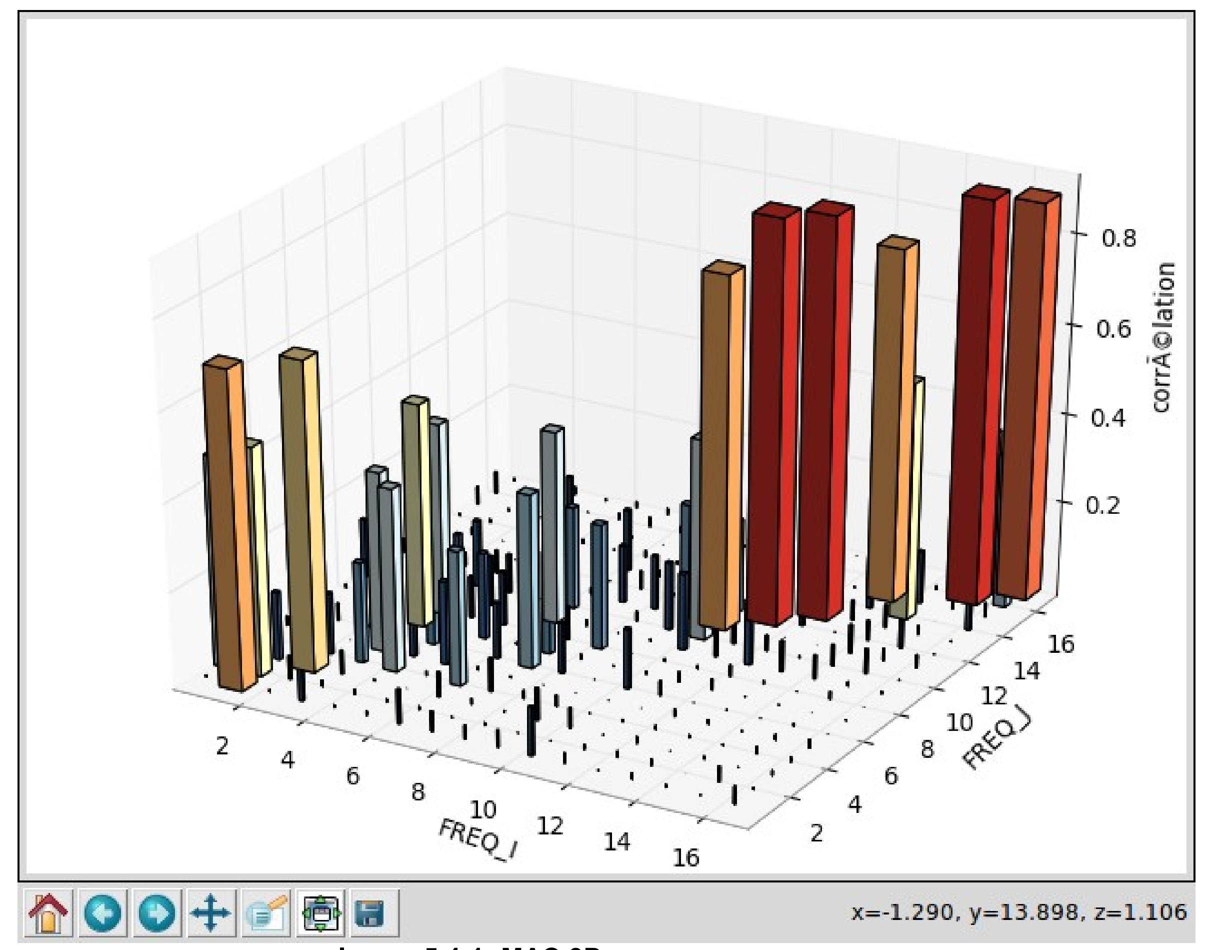


Image 5.4-1: MAC 3D.